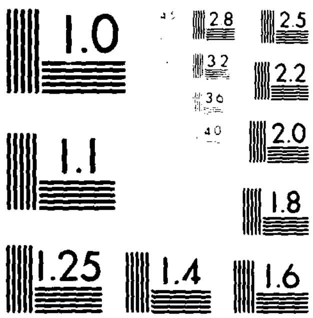
AD-A085 956 HONEYWELL SYSTEMS AND RESEARCH CENTER MINNEAPOLIS MN F/G 12/1
DECENTRALIZED CONTROL.(U)
MAY 80 E D JENSEN N00014-78-C-0511
UNCLASSIFIED 80SRC36 NL

1 OF 1
AD
A085956

END
DATE
FILMED
8-80
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

REPORT ONR

DECENTRALIZED CONTROL

E.D. JENSEN

HONEYWELL
SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

CONTRACT N00014-78-C-0511

MAY 1980

FINAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

PREPARED FOR THE
OFFICE OF NAVAL RESEARCH • 800 N. QUINCY ST. • ARLINGTON, VA. 22217

CHANGE OF ADDRESS

Organizations receiving reports on the initial distribution list should confirm correct address. This list is located at the end of the report. Any change of address or distribution should be conveyed to the Office of Naval Research, Code 211, Arlington, VA 22217.

DISPOSITION

When this report is no longer needed, it may be transmitted to other organizations. Do not return it to the originator or the monitoring office.

DISCLAIMER

The findings and conclusions contained in this report are not to be construed as an official Department of Defense or Military Department position unless so designated by other official documents.

REPRODUCTION

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A085956	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DECENTRALIZED CONTROL.		5. TYPE OF REPORT & PERIOD COVERED Final Report 7/1/78-12/30/79
		6. PERFORMING ORG. REPORT NUMBER 80SRC36
7. AUTHOR(s) E.D. Jensen		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0511
9. PERFORMING ORGANIZATION NAME AND ADDRESS Honeywell Systems and Research Center, ~~2600 Ridgway Parkway~~ Minneapolis, Minnesota 55413		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Street Arlington, Virginia 22217		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

9 Final rpt 1 Jul 78-30 Dec 79,

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Distributed computer systems
Operating systems
Decentralized control
Multilateral management
Communication

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A conceptual model of decentralized control for computer systems is presented. The model is founded on the principle that an activity at any particular level of abstraction is decentralized if there is no unique lower-level entity which enforces a consistent view of the activity state on the entities involved in that activity. Three factors are presented to determine how decentralized the control of an individual resource is: the number of controllers of a resource, the extent that each controller is involved in

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 402349

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

every control activity, and the parity of the controllers' authority. Two factors are presented to determine the decentralization of system-wide control: the number of controllers involved in each instance of multilateral management and the number of resources involved in each instance of multilateral management. Physical communication issues are discussed which effect the logical decentralization of control. Communication is considered to consist of the production and manifestation of signals. The observability of signals is the practical difference between the signal production and manifestation. Three factors of signal observability are presented: completeness, coherence, and latency.

Accession For	
NTIS GRA&I	☑
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

ACKNOWLEDGEMENT

This work has benefitted substantially from the comments of my colleagues at the Honeywell Systems and Research Center, Carnegie-Mellon University, and elsewhere.

CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

INTRODUCTION

Computer systems are usually considered to be "distributed" on the basis of such aspects as user access, system geography, processing, or data being decentralized. However, it appears to some researchers[1,2] that the most technologically interesting, and in many cases potentially valuable, aspect to decentralize is the system control. Unfortunately, there is little commonality of view on what "decentralized" (or even "centralized") control means. This is primarily because the more decentralized alternatives are only recently beginning to be perceived in any sort of conceptual fashion. Some degree of decentralized control has arisen in various aspects of computer system design, but almost inevitably out of convenience or necessity rather than through consideration of fundamental principles. The initial scarcity of processor resources focused the attention of system software designers on uniprocessors, where they developed the current foundations of traditional operating systems. As processor hardware became less costly, multiple processors were connected to shared primary memory ("multiprocessors"), and most of the uniprocessor software concepts and structures could be successfully retained. One consequence of this historical development was that many of the premises on which these traditional operating

---

[1] E. Douglas Jensen, "The Honeywell Experimental Distributed Processor--An Overview," Computer, January 1978.

[2] Philip H. Enslow, "What is a Distributed Processing System?," Computer, January 1978.

system concepts were strongly based are now very often so taken for granted that they have become transparent: they are either not recognized explicitly, or believed to be universally valid. This makes it difficult to see where they may be inherently centralized and how more decentralized alternatives might replace them.

To help remedy this, we present a model in which there is a spectrum of control decentralization from minimal (centralized) to maximal. We have three objectives for this model:

- Contribute to an improved understanding of the fundamental nature of "decentralization," particularly with respect to control
- Assist in the formulation of a common frame of reference and terminology for discussing control decentralization
- Facilitate the perception of relative differences and similarities among specific instances of control (although the model is not intended to provide quantitative evaluations)

The model is uninterpreted in the sense that it does not attempt to ascribe attributes (for example, "better," "more fault tolerant") to the points in the control spectrum. As this understanding, terminology, and perception sufficiently improve, it will become increasingly feasible to learn the implications of decentralized control; that is, to determine the application conditions under which various degrees of control decentralization result in what potential advantages and disadvantages.

The primitive object in this model is a resource, which is a type (that is, set of operations which collectively define its behavior). Resources exist at different levels of abstraction; in computer systems these tend to range from the user interface at the top, to the hardware ISP* at the bottom. (A type is implementation-independent, so hardware per se is not a level of abstraction.) The resources at the lowest level of interest are usually considered to be "real" (for example, storage locations) and those above to be "abstract" (for example, files). A resource is encapsulated by one or more controllers (also types) which abstract it and thus themselves become resources at the next higher level. This abstraction we call control; it consists of the decisions and actions involved in managing (for example, assigning, releasing, sharing) the resource.

In this model, the degree of control decentralization is based on the idea of multilateral resource management: the nature and extent of multiple controller involvement in the management activities for each resource individually and for all resources collectively. The model also considers the factors which limit the range of control decentralization feasible, or possible, in any particular situation; these are properties of the communication among the different controllers and resources.

We have formulated the model in geometric terms: the factors which determine the degree of control decentralization are considered to be the edges of one multi-dimensional construction which bounds a "design decision

*Instruction Set Processor [Bell and Newell, 1971]

space"; the factors which restrict the design decisions are the edges of another which bounds a "design constraint space." This formulation suggests characterizing each construction according to such properties as vertex identification (the extreme cases of each factor), edge (factor) metric, and edge orthogonality (independence of factors). Each of these constructions is discussed next.

SECTION 2

THE DESIGN DECISION SPACE

In this model, five major factors determine the degree of control decentralization. The first three deal with control of individual resources; the last two with control of the resources collectively.

INDIVIDUAL RESOURCE CONTROL

Individual resource control is fundamental, because it exists for every resource (whereas some resources may be managed only individually and not collectively). The degree to which the control of a single resource is decentralized is a function of the number of controllers it has, and of the relationships among them.

There are many different ways in which multiple controllers may all participate in the management of the same resource. For example:

- Successive, where all management is performed by one controller at a time
- Partitioned, where each controller performs a different part of the management (whether consecutively or concurrently)
- Democratic, where all controllers perform each management activity by negotiation and consensus among equals

The various forms of multilateral management exhibit different degrees of decentralization. This model distinguishes them according to two factors: concurrency and equipollence.

In the sense intended here, concurrency is the extent to which each management activity for a particular resource is carried out by all its controllers together. As with other interpretations of the term, this type of concurrency may be either real (requiring multiple processors) or virtual (for example, multiprogramming). Distinct activities may have different degrees of concurrency. These values may be retained as a vector for use in forming multifactor decentralization measures, or combined to create a scalar metric for this factor as well as for multifactor use. One such metric is the average number of concurrently participating controllers, scaled by some system- or application-dependent activity "importance" weights such as frequency of occurrence. (However, one must resist the temptation to quantify the model beyond its intent and suitability; it is sufficient for our purposes to show that in principle an adequate measure can be produced.) In our view, any such metric should define the most centralized case to be when only one controller performs a particular instance of any management activity on the resource; the most decentralized case should be when every controller of that resource is involved in every instance of every activity. Applying this factor alone to the example forms of multilateral management above shows that successive is the most centralized, democratic is the most decentralized, and partitioned is in between.

Equipollence is the degree of equality with which management authority and responsibility are distributed across the multiple controllers of a resource. As for concurrency, this too may be different for the various management activities. Equipollence can be usefully represented with a three-dimensional graph on which the Z axis shows the authority and responsibility of each controller with respect to each activity for that resource (see Figure 1). The successive and democratic forms would each be depicted (as in Figure 2) by a horizontal plane $Z = k$ (the two forms are distinguished by the concurrency factor). An example of partitioning by function is illustrated in Figure 3 as the set of points which lie on the $Z = 0$ plane except for those on the $X = Y$ diagonal, whose $Z = k$. In the context of this representation a suitable scalar equipollence metric is average Z-axis gradient (considering weighting by activity importance such as frequency if desired). The maximally centralized case is when there is no gradient across all but one of the controllers (say, i), and maximal difference between that one and all the rest (that is, a two-level hierarchy of control which asymtotically approaches autocracy). This is shown in Figure 4 as the set of points lying on the plane $Z = 0$ except that those for which $X = i$ have $Z = k$. The maximally decentralized case (see Figure 2) is when there is no gradient across any of the controllers: each is equally capable of participating in all the management activities for the resource.

In Figure 5, these first two factors (concurrency and equipollence) determining the degree of control decentralization are represented as the orthogonal edges of a two-dimensional construction. One corner is the maximally centralized point (minimum concurrency and minimum equipollence), which in the limit is autocracy. Diagonally opposed to that corner is the

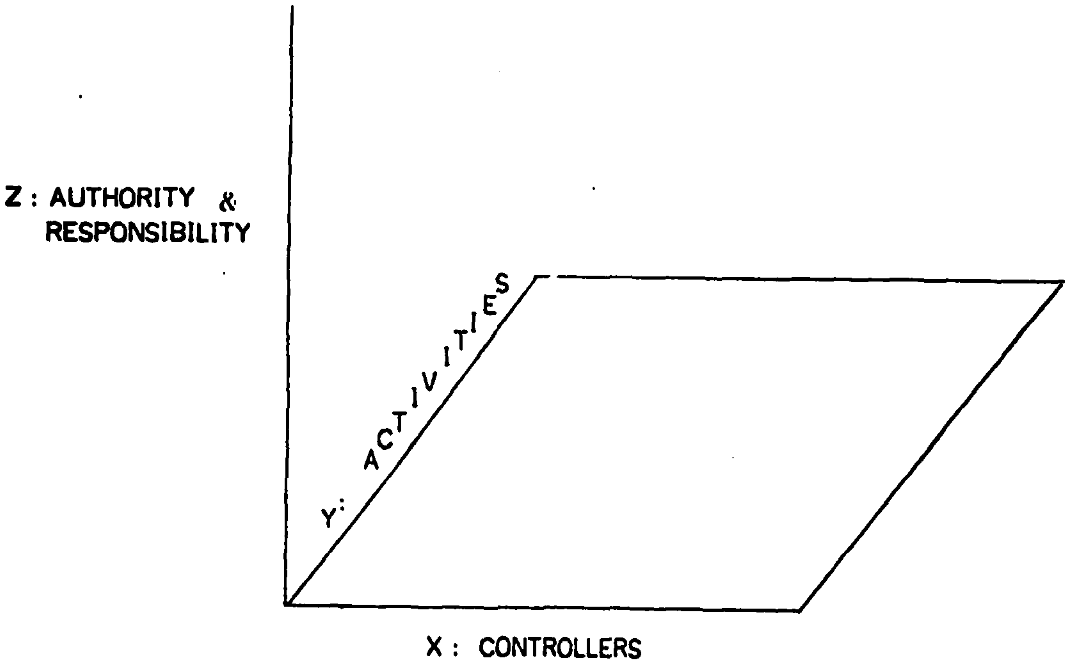


Figure 1. Graphical Representation of Equipollence

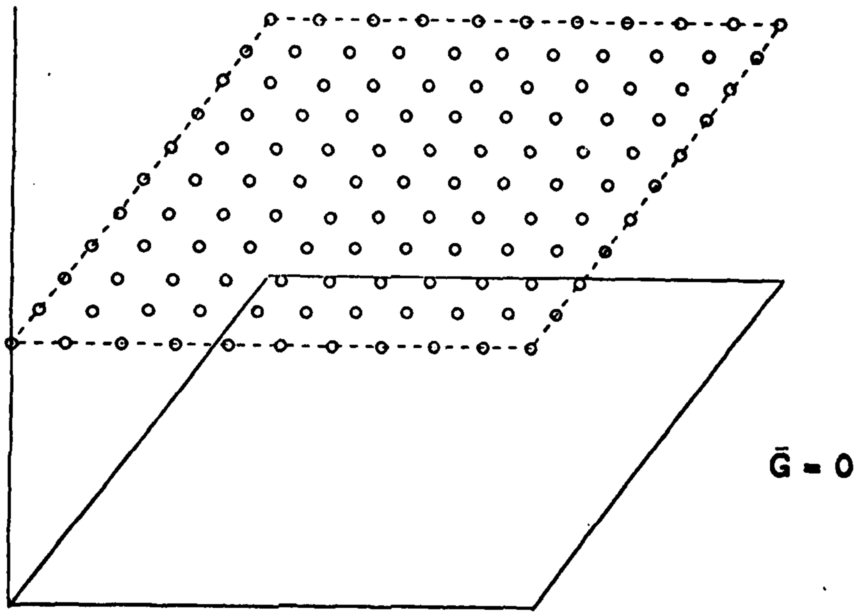


Figure 2. Successive and Democratic Forms of Control

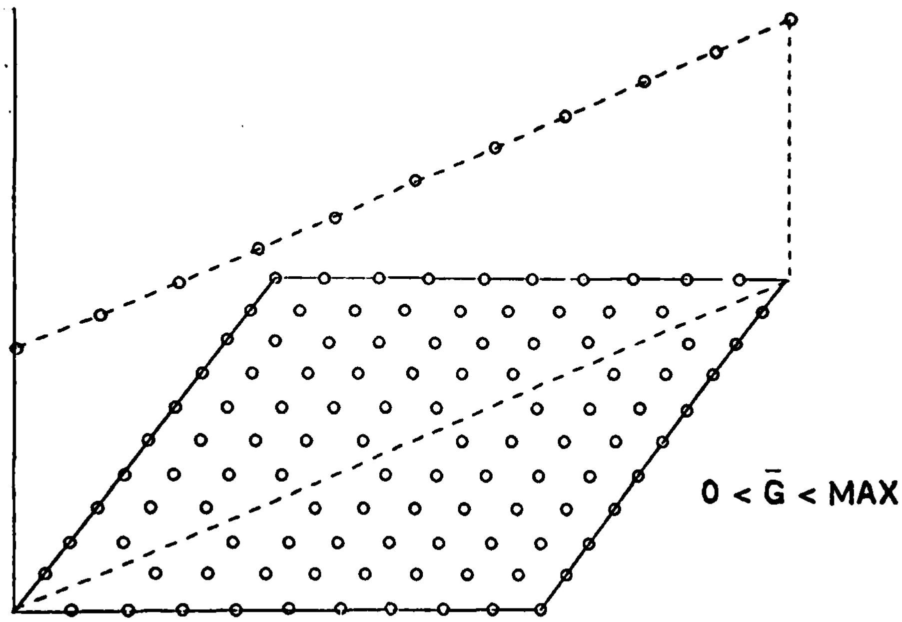


Figure 3. Example of Functional Partitioning

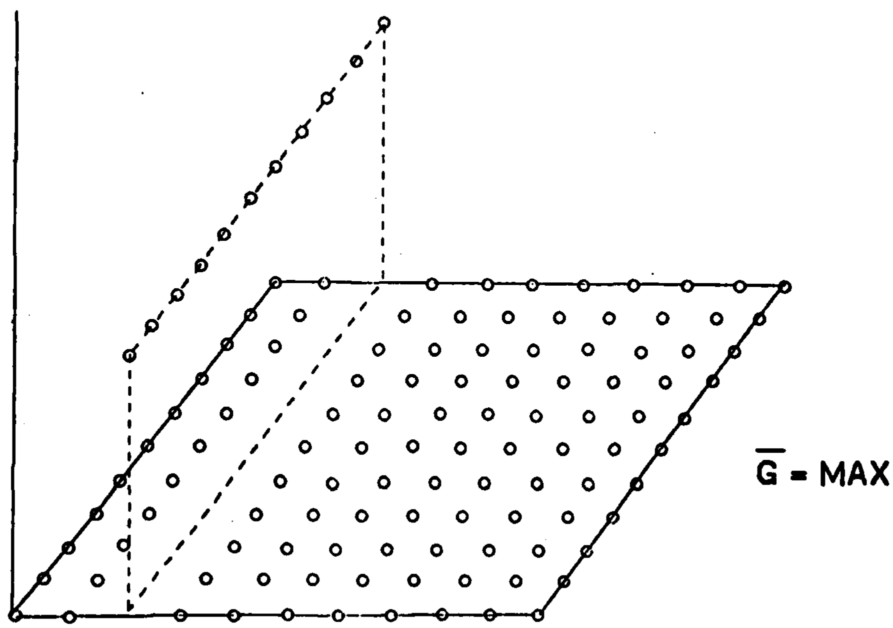


Figure 4. Maximally Centralized Case (two-level hierarchy asymtotically approaching autocracy)

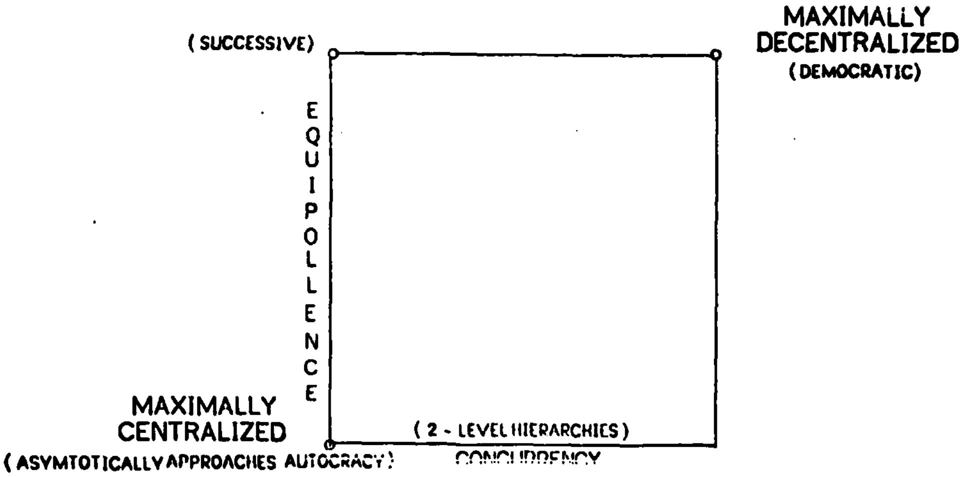


Figure 5. Graphical Representation of the First Two Factors

maximally decentralized point (maximum concurrency and maximum equipollence), exemplified by democracy. Successive is an intermediate case, where equipollence is high but concurrency is low. All the cases along the bottom concurrency edge are two-level (that is, maximum average gradient) hierarchies which differ from one another according to degree of concurrency; the maximum concurrency/minimum equipollence corner is more difficult than the others to associate with an obvious multilateral management technique.

The third factor contributing to the degree of control decentralization is the number of controllers a resource has. In general, this edge is orthogonal to the first two since neither concurrency nor equipollence are

affected by the number of participants. For example, the democratic and successive vertices of Figure 5 become edges in Figure 6. However, the most centralized endpoint of the third factor is an exception, because when there is only one controller management is then not multilateral, so concurrency and equipollence are both necessarily zero. In the geometric representation of our model we would say that the X = Y = Z = 0 corner is the only one which exists on the XZ plane. Control can be decentralized without limit (in principle) on this edge; it does not have a unique maximally decentralized endpoint.

Figure 6 also illustrates that while the minimally and maximally decentralized cases of individual resource control can be readily identified, most points within the construction are more difficult to order solely on the basis of their three coordinates. The function that determines this ordering must

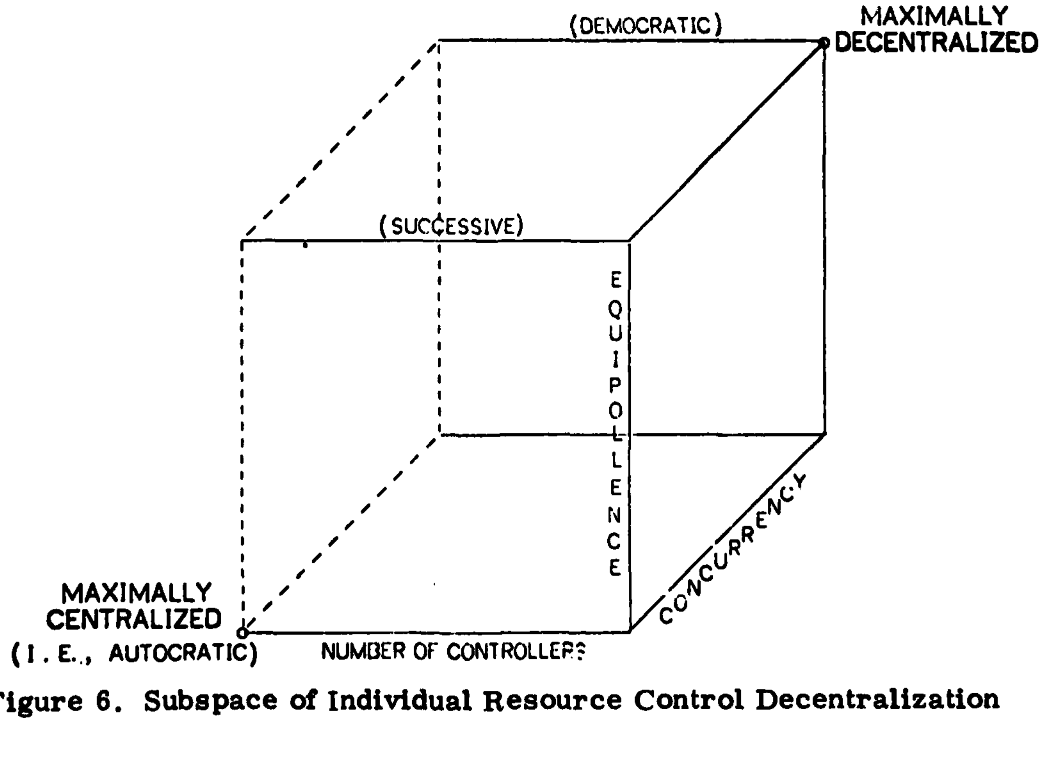


Figure 6. Subspace of Individual Resource Control Decentralization

also take into account the relative significance of each factor, which depends on the motivations and requirements of a specific system and application. For the purposes of this model, it currently appears sufficient to use a linear function of the three factors, the coefficients being their relative significances. A better understanding of how each factor relates to various system attributes may more clearly illuminate this issue.

COLLECTIVE RESOURCE CONTROL

Very often resources are controlled not just as separate entities but also collectively in accordance with more global objectives and constraints (that is, as a system). The three factors above do not account for the extent to which this latter aspect of control is decentralized, even by combining the per-resource results. The precept of multilateral resource management can be applied at the collective level to derive two factors that determine the degree to which system-wide control is decentralized.

The first of these factors (the fourth in our model) is the number of controllers involved in each instance of multilateral management. A scalar metric for this is the average percentage of all other controllers in the system with which each controller performs multilateral management. We combine the per-controller percentages with unity weighting because controllers (unlike resources) generally appear to have uniform importance with respect to the degree of decentralization. The maximally centralized case is when no controller participates in the multilateral management of any resource: the resources are partitioned into disjoint subsets, each of which is managed independently of the others by one controller. The

maximally decentralized case is when every controller participates with every other in the multilateral management of at least one resource. Figure 7 illustrates five cases ordered in degree of control decentralization according to this factor.

The second system-wide factor (number five in the model) is the number of resources involved in each instance of multilateral management. The scalar metric which suggests itself is the percentage of all resources in the system which are managed by at least two controllers. The maximally centralized case is when no resources are multilaterally managed; the maximally decentralized case is when all resources are multilaterally managed. In Figure 8, five cases are shown ordered in degree of control decentralization on this basis.

Together, the fourth and fifth factors provide a measure of system-wide control decentralization, as seen in the construction of Figure 9. If no controllers multilaterally manage, then obviously no resources are multilaterally managed; likewise, if no resources are multilaterally managed, then no controllers multilaterally manage. Thus, $X = Y = 0$ is the only point that exists on the lines $X = 0$ and $Y = 0$. The maximally centralized point in the construction is that no controllers multilaterally manage any resources; diagonally opposed is the maximally decentralized point where every controller participates in the management of every resource. Beyond that, ordering of cases in the space depends on the relative importance ascribed to each of the two factors by the particular system and application.

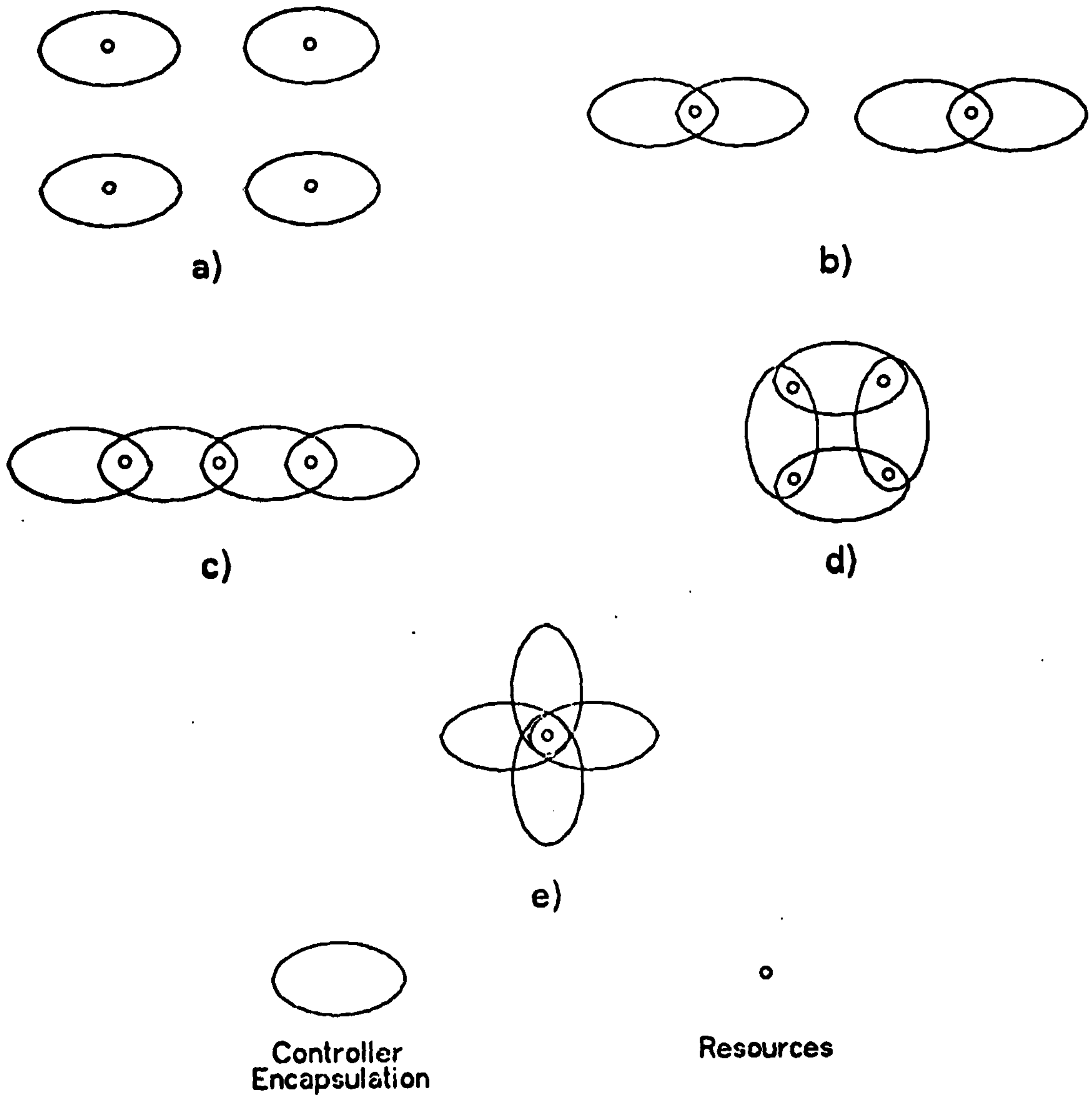


Figure 7. Ordering By Number of Controllers Which Participate in Each Instance of Multilateral Resource Management

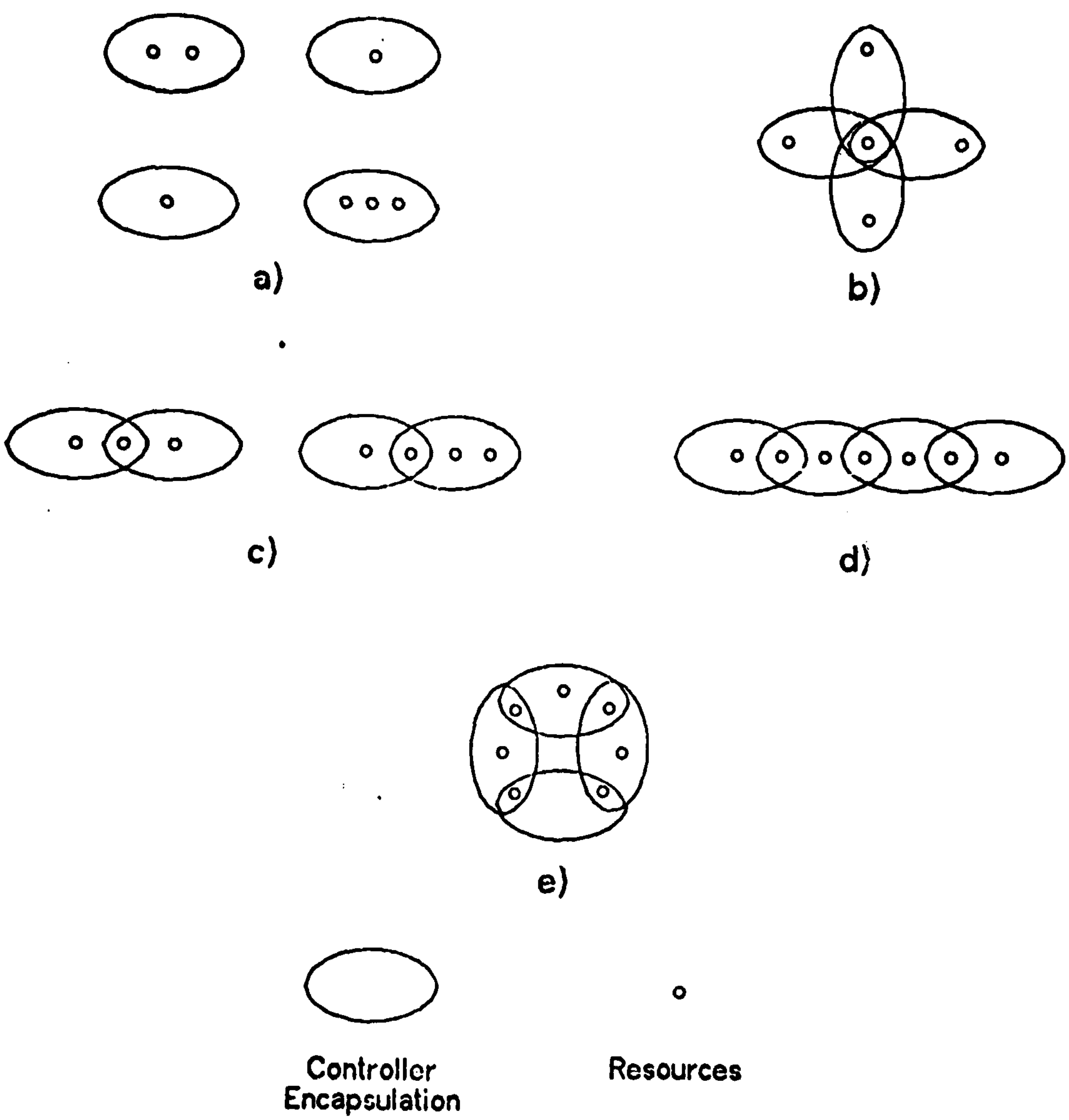


Figure 8. Ordering By Number of Multilaterlly Managed Resources

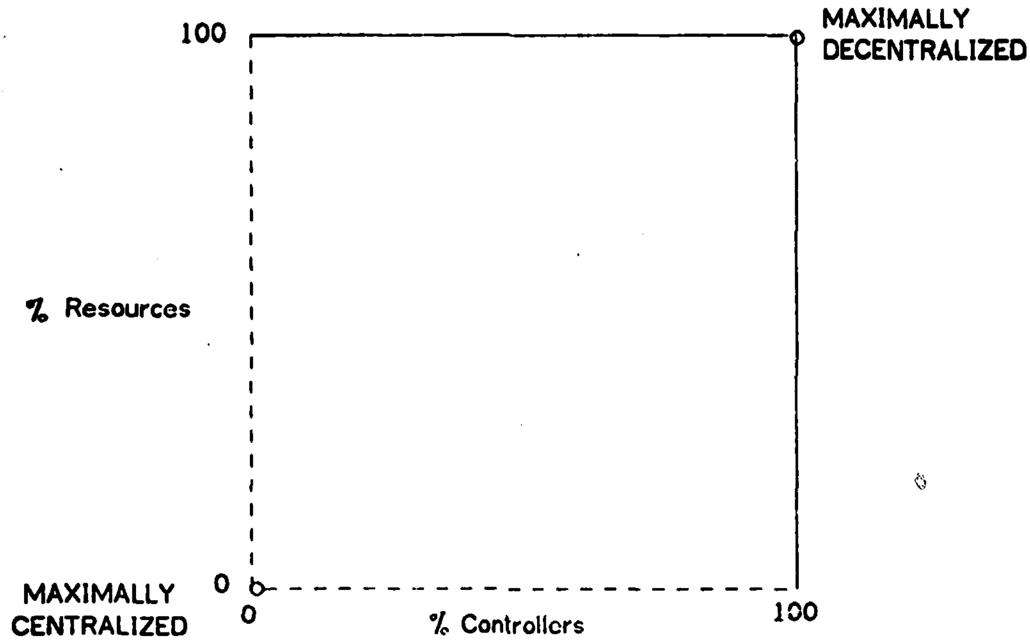


Figure 9. Subspace of System-Wide Control Decentralization

An interesting method for doing this is to perform a sum of products using the vector forms of the metrics. (While the vectors may not necessarily be the same length, there is an entry in that of factor five corresponding to every nonzero entry in that of factor four.) The cases in Figure 10 are in ascending order of system-wide control decentralization according to this measure.

We are unable to graphically depict the complete five-dimensional representation of the model. The edges corresponding to the fourth and fifth factors are orthogonal to the others but obvious boundary conditions do not exist; the maximally centralized collective case implies the maximally centralized individual case.

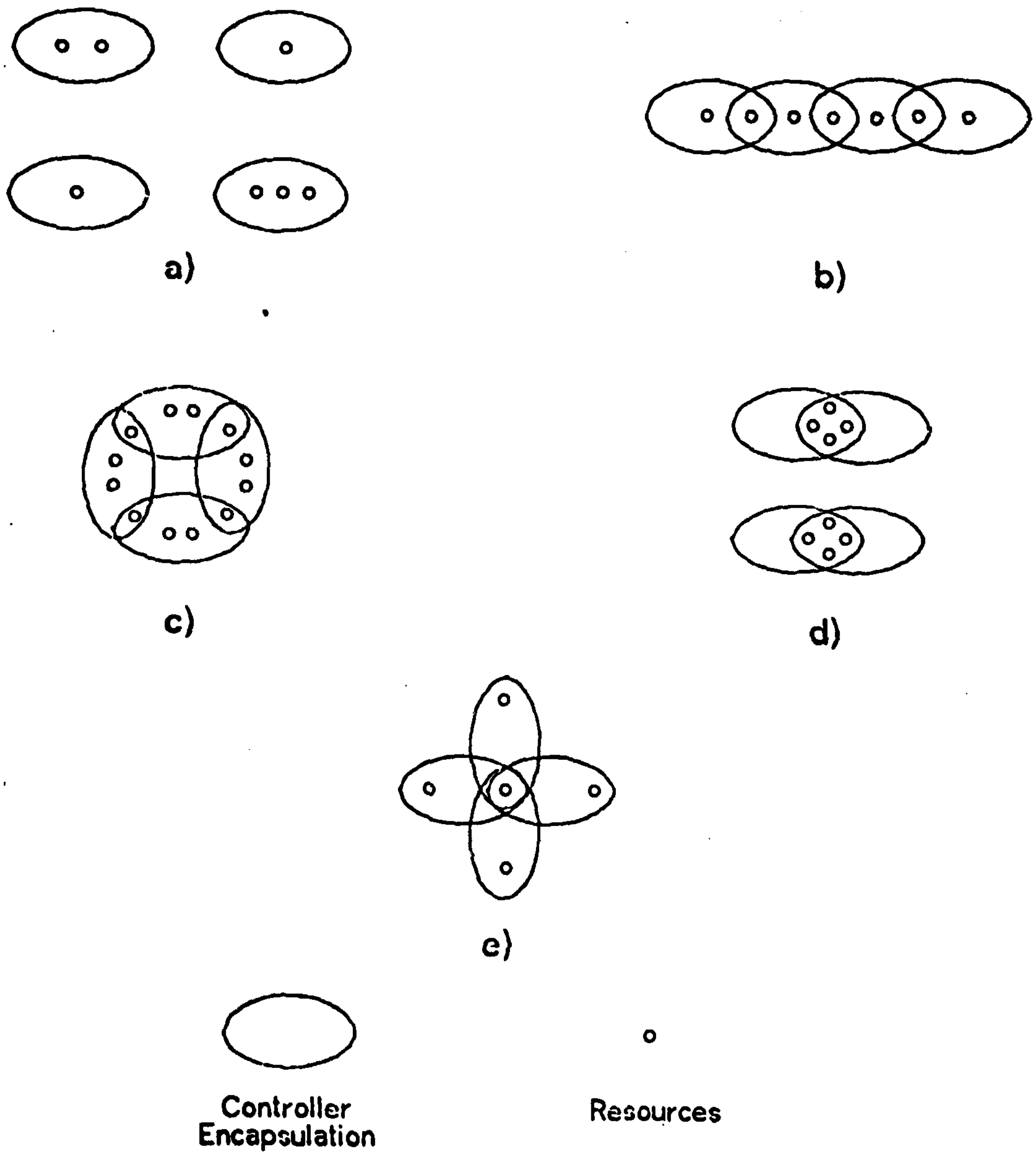


Figure 10. Ordering By Number of Resources Multilaterally Managed By Number of Controllers

There is a five-dimensional design decision space corresponding to any particular level of abstraction, and in general a computer system will not be represented at the same point in each. For example, a computer network could have:

- Rather decentralized control at the user interface level, provided by a so-called "network operating system."
- Rather centralized control at the executive level, because the host operating systems are autonomous.
- Rather decentralized control at the communication subnet level, as a consequence of both the hardware and the routing algorithm designs.

The implications of a system's representative point position at one level of abstraction on those at any other level are largely unknown at this time, especially for the more decentralized cases.

Normally designers do not have complete freedom to position a system anywhere desired in each space; a class of technical constraints imposes itself and limits the feasibility or even possibility of certain design options. These constraints are modeled in the three-dimensional design constraint space presented next.

SECTION 3

THE DESIGN CONSTRAINT SPACE

Multilateral resource management is strongly affected by the kinds and amounts of knowledge that the participating controllers have about each other. Some of this knowledge may be static (that is, an a priori model) and incorporate information about other controllers' strategies, tactics, and even algorithms. Other knowledge may be dynamic, including behavioristic models and current state of the other controllers. While static information is helpful in achieving a high degree of decentralized control, dynamic information is clearly essential. In our model of control, all dynamic information is represented as signals, which are the communications among controllers and among controllers and resources. Communication involves two conceptually distinct aspects of signalling: production and manifestation. The relationship between these two is termed signal observability. Signal observability has three important factors: completeness, latency, and coherence.

COMPLETENESS

Completeness of signal observability is the extent to which a controller can see any signal it wants to. More specifically, it is the probability for each controller that it can observe each signal in any particular set of signals. To more accurately model some cases, these probabilities may need to be conditional on certain aspects of the system state. A scalar measure of completeness can be obtained from the matrix of probabilities; what

usually matters is the probability values but not their matrix locations. The best (that is, least constraining) endpoint of this factor is that every controller can always observe every signal; the worst (that is, most constraining) endpoint is that no controller can ever observe any signal.

LATENCY

Latency of signal observability is the extent to which a controller can see a signal in time for it to be useful. More specifically, it is the probability for each controller that it can observe each signal in any particular set of signals (for example, those needed and which have precedence, processing time, or communication time constraints) any necessary amount of time before the next signal is sent (for example, in time to affect which signal is sent next). As with completeness the probabilities may need to be conditional and a scalar metric may be derived from the matrix. The best endpoint of this factor is that every controller can observe every signal within any arbitrarily small amount of time after it is sent; the worst endpoint is that no controller can observe any signal until after all signals in the set have been sent.

COHERENCE

Coherence of signal observability is the extent to which all controllers can have the same view of the system. More specifically, it is the probability for each controller in any particular set of controllers that it can induce the same ordering as they can on any particular set of signals (and thus have the same perception of any subset of the system state). Depending on the circumstances, consistency of signal ordering may be sufficient, or it

may be necessary for all controllers to observe the "actual" (with respect to a hypothetical global time reference) sending times in each observer's time reference. Again the probabilities may be conditional and a scalar metric can be derived. The best endpoint of this factor is that every controller can determine whatever consistent ordering is desired on all signals; the worst endpoint is that no controllers can determine any consistent ordering of any signals.

These three signal observability factors can be viewed as a three-dimensional construction enclosing a constraint space; separate spaces apply to different levels of abstraction in computer systems. As with the design decision space, the representative points are generally at different locations in each space. However, in this case more is known about the implications of a system's position at one level on its position at other levels; signal observability at any level typically can be no better than that at the level below it, because communications at one level are carried out by the next level down. Thus, signal observability at all levels ultimately rests on that at the lowest system-wide level: the processor communication hardware level.

The processor communication hardware level allows a processor to communicate with itself and any other processors in the system; it may be memory or an I/O mechanism such as a bus. Signal observability at this level is determined by many aspects, including: path topology and processor connectivity; intermediaries (routing and storage); transmission

times (path lengths and bandwidths); communication volume and priorities; initiation latencies (path and buffer allocation, processor multiplexing); errors and recovery. In particular, incoherence normally results from communication delays which are variable and unknown.[3]

---

[3]Gerard LeLann, "Distributed Systems--Toward A Formal Approach," Proc. IFIP Information Processing Congress, 1977.

SECTION 4

SIGNAL OBSERVABILITY AND DECENTRALIZED CONTROL

In uniprocessor and multiprocessor (that is, physically tightly coupled) systems, the shared main memory allows any process domain intersections (which leads to the need for protective restrictions). Consequently, the executive design is almost always based on the premise that a high degree of signal observability can be achieved at low cost. But in distributed (that is, physically loosely coupled) systems there is no shared main memory so there is a nontrivial, often very large, cost to achieve a high degree of signal observability (if it is even feasible at all).

Therefore, it is not currently possible for most distributed systems to have an operating system in the same sense as a uniprocessor or multiprocessor; that is, one that attempts to manage all the executive level resources as optimally as possible with respect to the best interests of the whole system. Instead, a typical distributed system is constrained by the state of the operating system art to being a <u>network</u> rather than a <u>computer</u>. The distinction is that a network has a separate operating system for each processor: the executive level resources are partitioned and each partition is managed locally for the good of just that small piece of the system. However, having only local and no global executive control severely restricts the nature and extent of processor interaction; for example, to resource sharing as opposed to multilateral resource management. For many applications (such as resource-sharing networks), such an arrangement

may be adquate or even necessary (for technical or non-technical reasons). But for many others (such as mission-oriented networks and real-time control), this greatly inhibits the extent to which certain important attributes, such as fault tolerance and modularity, can be provided on a system-wide basis.

Achieving a higher degree of global executive control than is presently attainable on a distributed system requires movement into the more decentralized regions of the design decision space. The executive must have no centralized data, procedure, clock, tokens, or hardware, and must have no hierarchical control relationships among the processors. Instead, it consists of a multiplicity of executive instantiations (one per processor) acting collectively to form a conceptually singular executive for the whole system. Because of the interprocessor communication problems characterized in the design constraint space, decentralized resource management algorithms must often differ dramatically in another way from their more centralized counterparts: they must make "best effort" decisions based on probablistically accurate and incomplete information (just as human managers do).

We call a loosely coupled computer having a high degree of decentralized system-wide executive control a distributed computer system. This type of system differs from:

- An I/O multiplex bus system such as MIL-STD-1553B,[4] which is intended to interface a small number of processors to a large number of I/O devices, whereas a distributed computer has processors interconnected for cooperative execution

[4] USAF, MIL-STD-1553: Military Standard Aircraft Internal Time Division Multiplex Bus, 1973.

- A local computer network such as Ethernet,[5] which is a collection of separate but interconnected computer systems, whereas a distributed computer has the system-wide executive control necessary to integrate the multiplicity of processors into a single computer

- A multiprocessor such as C.mmp,[6] which has centralized system-wide executive control, whereas a distributed computer has decentralized system-wide executive control. Cm*[7] under either the StarOS[8] or Medusa[9] operating systems is an intermediate case with partitioned executive control.

Some degree of decentralized system-wide control has been achieved at levels above and below the executive level. For example:

- At the application level in some multicomputers for real-time control, and in some "mission-oriented" computer networks for transaction and military uses

- At the communication subnet level in some computer networks

---

[5] Robert M. Metcalfe, "Ethernet: Distributed Packet Switching for Local Computer Networks," <u>CACM</u>, July 1976.

[6] William A. Wulf, and Gordon C. Bell, "C.mmp--A Multi-Processor," <u>Proc. AFIPS FJCC</u>, 1972.

[7] Richard J. Swan, "Cm*--A Modular Multi-Microprocessor," <u>Proc. AFIPS NCC</u>, 1977.

[8] Anita K. Jones, "Software Management of Cm*--A Distributed Multi-processor," <u>Proc. AFIPS NCC</u>, 1977.

[9] John K. Ousterhout, Donald A. Scelza, and Pradeep S. Sindhu, "Medusa: An Experiment in Distributed Operating System Structure," <u>Comm. ACM</u>, Vol. 23, No. 2, February 1980, pp. 92-105.

However, distributed computer systems are just beginning to emerge in research laboratories[1,10,11] because the control functions are more general and complex and the resources are more abstract and dynamic at the executive level than at these other levels. In our view, successfully achieving a distributed computer system will require not only new concepts and techniques of control, but also corresponding (and probably unconventional) new insights into hardware/software tradeoffs. Executive control will have to be considered a system problem, not just a software problem.

---

[10] Earl W. Boebert, William R. Franta, Douglas E. Jensen, and Richard Y. Kain, "Design Issues in A Distributed Executive," Proc. IEEE Compsac, 1978.

[11] Earl W. Boebert, William R. Franta, Douglas E. Jensen, and Richard Y. Kain, "Kernal Primitives of the HXDP Executive," Proc. IEEE Compsac, 1978.

REFERENCES

1. Jensen, E. Douglas, "The Honeywell Experimental Distributed Processor--An Overview," Computer, January 1978.

2. Enslow, Philip H., "What is a Distributed Processing System?" Computer, January 1978.

3. LeLann, Gerard, "Distributed Systems--Toward A Formal Approach," Proc. IFIP Information Processing Congress, 1977.

4. USAF, MIL-STD-1553: Military Standard Aircraft Internal Time Division Multiplex Bus, 1973.

5. Metcalfe, Robert M., "Ethernet: Distributed Packet Switching for Local Computer Networks," CACM, July 1976.

6. Wulf, William A., and Bell, C. Gordon, "C.mmp--A Multi-Processor," Proc. AFIPS FJCC, 1972.

7. Swan, Richard J., "Cm*--A Modular Multi-Microprocessor," Proc. AFIPS NCC, 1977.

8. Jones, Anita K., "Software Management of Cm*--A Distributed Multiprocessor," Proc. AFIPS NCC, 1977.

9. Ousterhout, John K., Scelza, Donald A., Sindhu, Pradeep S., "Medusa: An Experiment in Distributed Operating System Structure," Comm. ACM, Vol. 23, No. 2, February 1980, pp. 92-105.

10. Boebert, W. Earl, Franta, William R., Jensen, E. Douglas, and Kain, Richard Y., "Design Issues in A Distributed Executive," Proc. IEEE Compsac, 1978.

11. Boebert, W. Earl, Franta, William R., Jensen E. Douglas, and Kain, Richard Y., "Kernel Primitives of the HXDP Executive," Proc. IEEE Compsac, 1978.


8